Applied Mathematical Sciences, Vol. 7, 2013, no. 24, 1195 - 1197
HIKARI Ltd, www.m-hikari.com

On Reoptimization of the Shortest Common Superstring Problem

Vladimir Popov

Department of Intelligent Systems and Robotics
Ural Federal University
620083 Ekaterinburg, Russia
Vladimir.Popov@usu.ru

Abstract

In general, a reoptimization gives us a possibility to obtain a solution for a larger instance from a solution for a smaller instance. In this paper, we consider a possibility of usage of a reoptimization to solve the shortest common superstring problem.

Keywords: reoptimization, common superstring, Hamming distance

Investigation of different regularities has become essential in modern computer science (see e.g. [1] – [7]). In this paper, we consider the shortest common superstring problem.

The Shortest Common Superstring Problem (SCS):

Instance: *A collection $\mathcal{S}$ of strings over Σ and a positive integer k.*

Question: *Is there a string S such that $|S| \leq k$ and S is a superstring of $\mathcal{S}$?*

The problem SCS is **NP**-complete [8]. Moreover, SCS is **MAX SNP**-hard [9]. The following problem was proposed in [10].

The Shortest Common Superstring Reoptimization Problem with Adding a String (SCS+):

Instance: *A substring-free collection $\mathcal{S}$ of strings over Σ, an optimal solution of* SCS *for $\mathcal{S}$, and a string $T \notin \mathcal{S}$ such that $\mathcal{S} \cup \{T\}$ is substring-free.*

Task: *Find an optimal solution of* SCS *for $\mathcal{S} \cup \{T\}$.*

We can try to use SCS+ for solution of SCS. However, SCS+ is **NP**-hard [10]. We need some efficient algorithm for SCS+. But, there is no clear evidence for usage of an optimal solution of SCS for $\mathcal{S}$ for finding an optimal solution of SCS for $\mathcal{S} \cup \{T\}$.

Note that if $\mathcal{S} = \{ab^2, c^2a\}$, then c^2ab^2 is the shortest common superstring for $\mathcal{S}$. If $T = ca^2b$, then $c^2a^2b^2$ is the shortest common superstring for $\mathcal{S} \cup \{T\}$. So, we can try to use an optimal solution of SCS for $\mathcal{S}$ to solve SCS for $\mathcal{S} \cup \{T\}$ under assumption that an optimal solution of SCS for $\mathcal{S}$ is a subsequence of an optimal solution for $\mathcal{S} \cup \{T\}$.

Proposition 1. *For any $n > 0$, there is a collection $\mathcal{S}$ and strings S, T, and W such that S is the shortest common superstring for $\mathcal{S}$, W is the shortest common superstring for $\mathcal{S} \cup \{T\}$, S is not a subsequence of W, and $|S| > n$.*

Proof. It is easy to check that we can consider $\mathcal{S} = \{a^nb^n, c^na^n\}$, $S = c^na^nb^n$, $T = b^nc^n$, $W = a^nb^nc^na^n$. □

Therefore, in general case, we can not extend a shortest common superstring for $\mathcal{S}$ to a shortest common superstring for $\mathcal{S} \cup \{T\}$.

Let $D_H(X, Y)$ be the Hamming distance between strings X and Y.

Proposition 2. *For any $n > 0$, there is a collection $\mathcal{S}$ and strings S, T, and W such that S is a shortest common superstring for $\mathcal{S}$, W is a shortest common superstring for $\mathcal{S}$, W is a shortest common superstring for $\mathcal{S} \cup \{T\}$, $|S| = |W|$, $D_H(S, W) = |W|$, and $|S| > n$.*

Proof. It is easy to check that we can consider $\mathcal{S} = \{a^nb^n, b^na^n\}$, $S = a^nb^na^n$, $T = ba^nb$, $W = b^na^nb^n$. □

So, in general case, knowledge of an optimal solution for $\mathcal{S}$ does not provide any significant advantage in finding of an optimal solution for $\mathcal{S} \cup \{T\}$. Therefore, the assumption that an optimal solution of SCS for $\mathcal{S}$ is a subsequence of an optimal solution for $\mathcal{S} \cup \{T\}$ allows us to solve SCS only in some special case. We consider the following version of SCS+.

The Shortest Common Superstring Reoptimization Problem with Addition (SCS-Ad):

Instance: *A substring-free collection $\mathcal{S}$ of strings over Σ, an superstring S of $\mathcal{S}$, a string $T \notin \mathcal{S}$ such that $\mathcal{S} \cup \{T\}$ is substring-free, and a positive integer k.*

Question: *Is there a string W such that $|W| \leq k$, W is a superstring of $\mathcal{S} \cup \{T\}$, and S is a subsequence of W.*

Proposition 3. SCS-Ad *can be solved in polynomial time.*

Proof. Let $\mathcal{S} = \{S_1, \ldots, S_n\}$. For simplicity, we use $X[i]$ to denote the ith letter in string X, and $X[i, j]$ to denote the substring of X consisting of the ith letter through the jth letter.

Let $S_i = S[u[i], v[i]]$, for any $1 \leq i \leq n$ and for some $1 \leq u[i] < v[i] \leq |S|$. Since $\mathcal{S}$ is substring-free, it is easy to see that $u[i] \neq u[j]$, for any $1 \leq i < j \leq n$. For any superstring S of $\mathcal{S}$, we can consider a permutation $\pi = (i[1], \ldots, i[n])$ on the set $\{1, \ldots, n\}$ such that $u[i[1]] < \ldots < u[i[n]]$. It is easy to see that if $v[i[j]] + 1 < u[i[j + 1]]$, for some $1 \leq j < n$, then $S[1, v[i[j]]]S[u[i[j + 1]], |S|]$ is a superstring of $\mathcal{S}$. Therefore, we can assume that $u[i[j + 1]] \leq v[i[j]] + 1$. We define overlap $o(X, Y)$ of strings X and Y by $\max\{p \mid X[i] = Y[j], 1 \leq$

$j \leq p, |X| - p + 1 \leq j \leq |X|\}$. Since $\mathcal{S}$ is substring-free, we can assume that $u[i[j+1]] = v[i[j]] + 1 - o(S_{i[j]}, S_{i[j+1]})$. So, each permutation π uniquely defines some superstring of $\mathcal{S}$. Moreover, for any shortest superstring of $\mathcal{S}$ there is a permutation π which defines this superstring. Therefore, we can consider a solution of SCS as a permutation. Since any solution of SCS can be represented by some permutation on the set $\{1, \ldots, n\}$, it is easy to see that there are only $n+1$ positions for T. □

ACKNOWLEDGEMENTS. The work was partially supported by Analytical Departmental Program "Developing the scientific potential of high school" 8.1616.2011.

References

[1] A. Gorbenko and V. Popov, Multiple Occurrences Shortest Common Superstring Problem, *Applied Mathematical Sciences*, 6 (2012), 6573-6576.

[2] A. Gorbenko and V. Popov, On the Longest Common Subsequence Problem, *Applied Mathematical Sciences*, 6 (2012), 5781-5787.

[3] A. Gorbenko and V. Popov, The set of parameterized k-covers problem, *Theoretical Computer Science*, 423 (2012), 19-24.

[4] V. Yu. Popov, Computational complexity of problems related to DNA sequencing by hybridization, *Doklady Mathematics*, 72 (2005), 642-644.

[5] V. Popov, The approximate period problem for DNA alphabet, *Theoretical Computer Science*, 304 (2003), 443-447.

[6] V. Popov, The Approximate Period Problem, *IAENG International Journal of Computer Science*, 36 (2009), 268-274.

[7] V. Popov, Multiple genome rearrangement by swaps and by element duplications, *Theoretical Computer Science*, 385 (2007), 115-126.

[8] J. Gallant, D. Maier, and J.A. Storer, On Finding Minimal Length Superstrings, *Journal oj Computer and System Sciences*, 20 (1980), 50-58.

[9] A. Blum, T. Jiang, M. Li, J. Tromp, and M. Yannakakis, Linear Approximation of Shortest Superstring, *Journal of the ACM*, 41 (1994), 630-647.

[10] D. Bilò, H.-J. Böckenhauer, D. Komm, R. Královič, T. Mömke, S. Seibert, and A. Zych, Reoptimization of the Shortest Common Superstring Problem, *Lecture Notes in Computer Science*, 5577 (2009), 78-91.

Received: December 3, 2012